# Symmetric Equations on the Surface of a Sphere

Gary L. Russell, David H. Rind, Jeffrey Jonas
NASA Goddard Institute for Space Studies

Published in *Geoscientific Model Development*
December 2018
doi:10.5194/gmd-2018-126

## Part I : Symmetric Equations on the Surface of a Sphere

Three symmetric coordinates used for two-dimensional horizontal flow.

Advantages:

All quantities are defined continuously over the whole spherical surface.

All three components of vector quantities use same lines of computer code.

## Part II : Icosahedral B-grid model programmed with Symmetric Equations

Primary horizontal grid cells are 12 regular pentagons and numerous irregular hexagons.

B-grid means centers of momentum cells coincide with corners of primary grid cells.

## Coordinates

According to the “hairy ball theorem” of Poincare, every continuous horizontal vector field on the surface of a sphere has a **0**.

A continuous unit vector on the spherical surface has a discontinuity.

A differentiable coordinate on the spherical surface requires a continuous unit vector which will have a discontinuity.

Latitude and longitude coordinates are discontinuous at north and south poles.

Also, if two coordinates are not orthogonal on a surface,
then greater obtuseness of the angle between the coordinates decreases stability and precision of the results.

Suppose the hairs are straight and are one unit long representing a coordinate which has a polar discontinuity.

Next, suppose the hairs represent horizontal velocity and are cut smoothly so that they are of length 0 at the pole.

Although the coordinate still has a discontinuity, velocity is continuous everywhere because it is **0** at the pole.

## Problem with two coordinate flow

For horizontal vector fields to be continuous they must be **0** where the coordinate is discontinuous.

For this reason, shallow water models were rewritten using derivatives of scalar quantities such as kinetic energy, vorticity, divergence, stream function, or velocity potential using local coordinates which may be different from from those of their neighbors. After manipulation, vector velocity is resurrected. Models are much more complicated.

To be shown later with 3 symmetric coordinates:

A derivative with respect to a coordinate or a vector field will continuously approach 0 as the coordinate approaches its pole. Consequently, quantities are continuous over the whole sphere.

At one coordinate pole, the other two coordinates are perpendicular

## Symmetric Coordinates on the Sphere

Center of the sphere is at the origin with orthogonal axes **X**, **Y**, and **Z**.
**X** and **Y** pass through Earth's equator; **Z** aligns with north-south axis.

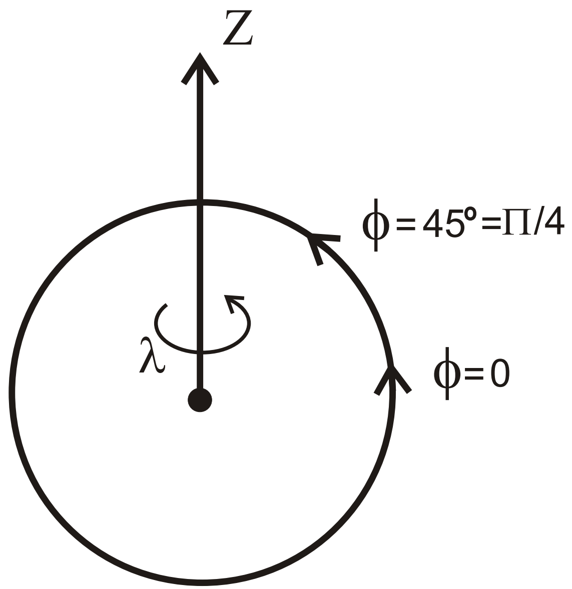
Two separate triplets of symmetric coordinates are defined:
$\mu$, $\nu$, $\lambda$ = longitude (asimuth) of the axes **X**, **Y**, **Z** respectively
$\delta$, $\varepsilon$, $\varphi$ = latitude (altitude) of the axes **X**, **Y**, **Z** respectively

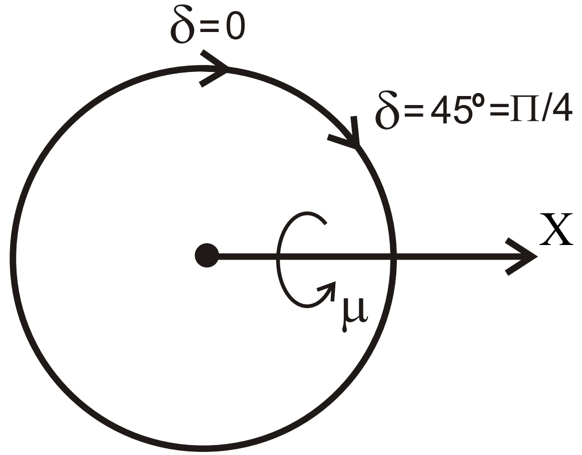
$\lambda$ and $\varphi$ are Earth longitude and latitude, **Z** is the north-south axis.

## New Coordinates

The **Z** axis, aligned with Earth's north-south axis, uses the Earth's labels: latitude, longitude, equator, poles. The **X** and **Y** axes want to use those labels also.

In symmetric equations, any mention of **Z** axis labels in an equation also occurs in a similar fashion to **X** and **Y** axes.



Position vector **P** on unit sphere

**P** = vector from center of sphere to point on unit sphere.

**P** has magnitude 1.

**P** is perpendicular to horizontal vectors on the surface of the sphere.

**P** = (x, y, z) =

$= (\sin\delta, \cos\mu \cos\delta, \sin\mu \cos\delta) =$

$= (\sin\nu \cos\varepsilon, \sin\varepsilon, \cos\nu \cos\varepsilon) =$

$= (\cos\lambda \cos\varphi, \sin\lambda \cos\varphi, \sin\varphi) =$ normal Lat and Lon

$= (\sin\delta, \sin\varepsilon, \sin\varphi) =$ symmetric coordinates

Extra credit: compute (x, y, z) as symmetric function of μ, ν, and λ

Gradient of scalar $h$ defined by symmetric and Lat-Lon coordinates:

$\nabla h = (\cos\delta\ \partial h/\partial\delta,\ \cos\varepsilon\ \partial h/\partial\varepsilon,\ \cos\varphi\ \partial h/\partial\varphi)\ /\ R$ Symmetric

$\nabla h = (\mathbf{E}\ \partial h/\partial\lambda + \mathbf{N}\ \cos\varphi\ \partial h/\partial\varphi)\ /\ R\cos\varphi$ Lat-Lon

$\nabla h$ is horizontal

$R$ = radius of sphere (m)

**E** and **N** = unit vectors in eastward and northward directions

Laplacian of scalar $h$ defined by symmetric and Lat-Lon coordinates:

$\nabla^2 h = (\partial^2 h/\partial\mu^2 + \partial^2 h/\partial\nu^2 + \partial^2 h/\partial\lambda^2)\ /\ R^2$ Symmetric

$\nabla^2 h = [\partial^2 h/\partial\lambda^2 + \cos\varphi\ \partial(\cos\varphi\ \partial h/\partial\varphi)/\partial\varphi]\ /\ R^2\cos^2\varphi$ Lat-Lon

When point approaches north or south pole, $\partial h/\partial\lambda$ approaches zero.

**A** = Specific Angular Momentum on unit Sphere (m/s)

**A** = **P** × **V**

**V** = **A** × **P**

**V** = horizontal velocity (m/s)

**A** is horizontal, has the same magnitude as **V**, and is at right angle to **V**.

Advecting **A** removes the metric term from the momentum equation.

Flux form advection conserves global **A** exactly.

**A** simplifies several formulas where local coordinates were formerly used.

Alignment, vectors should be horizontal

Specific angular momentum, **A**, is supposed to be horizontal, tangential to the spherical surface.

Vertical column mixing, either moist convection or vertical advection, maintains alignment; the Coriolis force maintains alignment.

Pressure gradient force via Green's theorem and horizontal advection distorts alignment; **A** is no longer horizontal.

Alignment is restored by projecting **A** onto the sphere's tangent plane.

$\mathbf{A}_{\mathrm{ALIGNED}} = \mathbf{A} - (\mathbf{P}\cdot\mathbf{A})\mathbf{P}$

Individual alignment errors are small.

Mass flux (kg/s) across a spherical arc from $\eta_1$ to $\eta_2$

$$M = \int_{\eta_2}^{\eta_1} Rh\, \mathbf{V} \cdot \mathbf{F}\, d\eta = \int_{\eta_2}^{\eta_1} Rh\, (\mathbf{P} \times \mathbf{V}) \cdot (\mathbf{P} \times \mathbf{F})\, d\eta = \int_{\eta_2}^{\eta_1} Rh\, \mathbf{A} \cdot \mathbf{E}\, d\eta =$$

$$= R \int_{\eta_2}^{\eta_1} h\, \mathbf{A} \cdot d\mathbf{P}$$

$h$ = mass per unit area ($kg/m^2$)

$\mathbf{F}$ = horizontal unit vector perpendicular to arc

$\mathbf{E}$ = horizontal unit vector parallel to arc

$M_{n-1/2} = R\, h_{n-1/2}\, \mathbf{A}_{n-1/2}\, (\mathbf{P}_n - \mathbf{P}_{n-1})$ = finite difference mass flux

Integrating over the grid cell corner points: $\mathbf{P}_0, \mathbf{P}_1, .. \mathbf{P}_N = \mathbf{P}_0$

and setting grid cell area to $K$:

$\Delta h = \Delta t\, \Sigma\, M_{n-1/2} / K = R\, \Delta t\, \Sigma\, h_{n-1/2}\, \mathbf{A}_{n-1/2}\, (\mathbf{P}_n - \mathbf{P}_{n-1}) / K$

Similarly, the <u>Pressure Gradient Force</u> for shallow water equations is:

$\Delta \mathbf{A} = R\, \Delta t\, \Sigma\, \mathbf{\Phi}_{n-1/2}\, (\mathbf{P}_n - \mathbf{P}_{n-1}) \,/\, K$

$\Delta h = R\, \Delta t\, \Sigma\, h_{n-1/2}\, \mathbf{A}_{n-1/2}\, (\mathbf{P}_n - \mathbf{P}_{n-1}) \,/\, K$    repeated for comparison

$\mathbf{\Phi} = h$ * Gravity = geopotential ($m^2/s^2$)

For $\Delta \mathbf{A}$, **P** loops over the corners of the momentum A or B cells.

**P** and **Φ** are centered at primary C cells, $\mathbf{\Phi}_{n-1/2}$ is an average value.

It is also possible to assume that **Φ** is linearly interpolated from corners n-1 to n and compute the integrand for an edge in closed form.

For $\Delta h$, **P** loops over the corners of the primary C cells.

Since Williamson et al. [1992] numerical schemes for the shallow water equations have been compared using Rossby-Haurwitz wave 4.

For cubed-sphere models starting from Rossby-Haurwitz wave 4, each wave-length remains identical because it lies above the same grid arrangement.

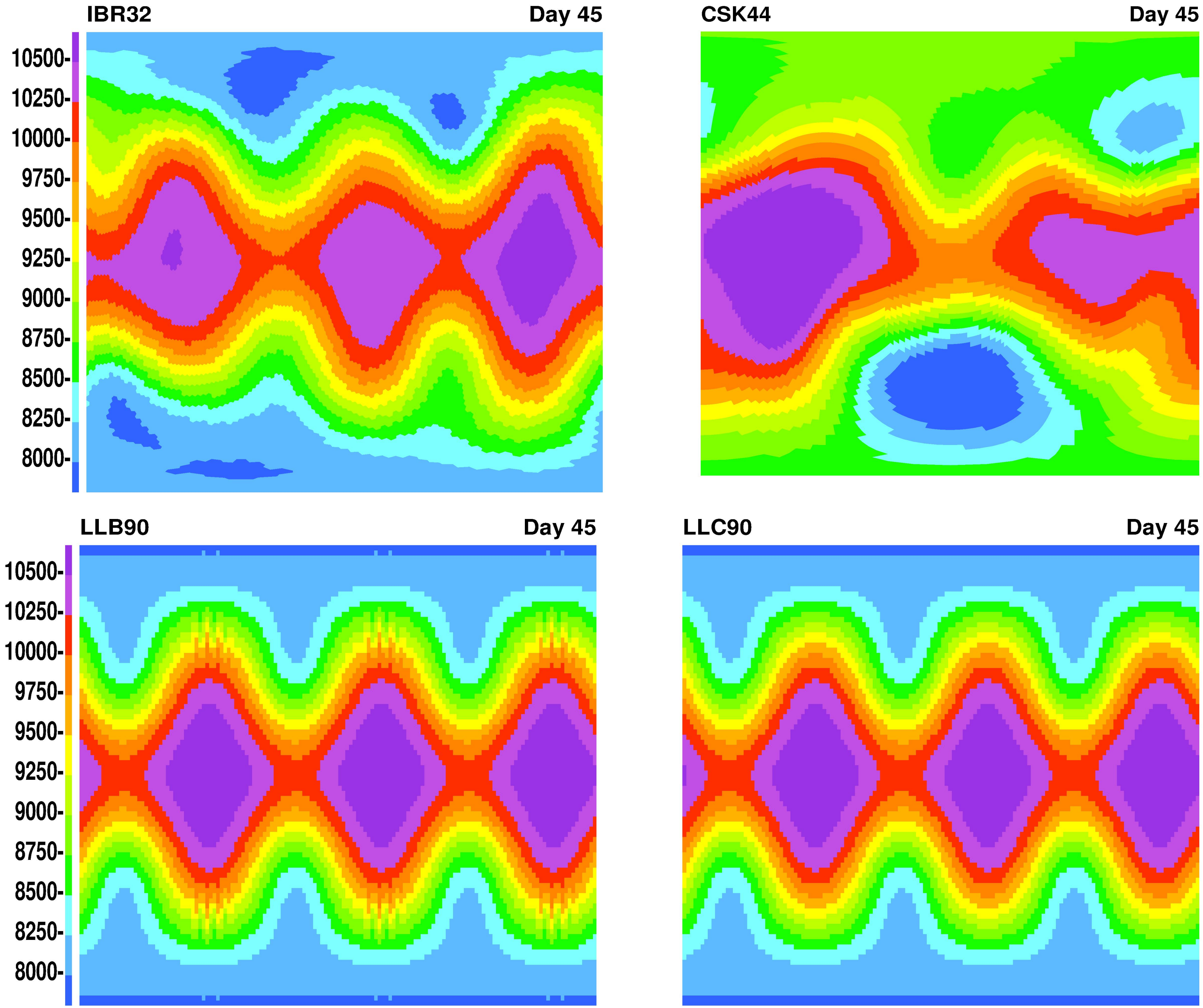
For icosahedral models starting from Rossby-Haurwitz wave 5, each wave-lengths remains identical.

3 is relatively prime to both 4 and 5, so we use Rossby-Haurwitz wave 3.

With RH wave 3, separate wave-lengths diverge among themselves, and there is more variety in the errors that may occur.

Icosahedral model decays on day 45, cubed-sphere decays on day 26; each of these two model use symmetric equations.

Rossby-Haurwitz wave 3.
Shallow water equations.
Both icosahedral and cubed-sphere grid use symmetric equations.
Arakawa B-grid and C-grid are Lat-Lon schemes.



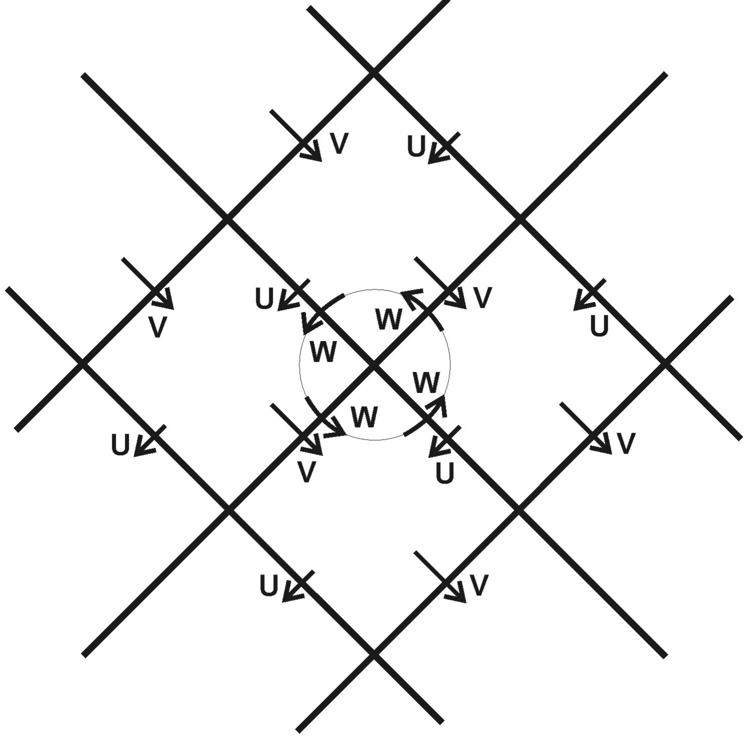
Cubed-Sphere grid cell arrangement

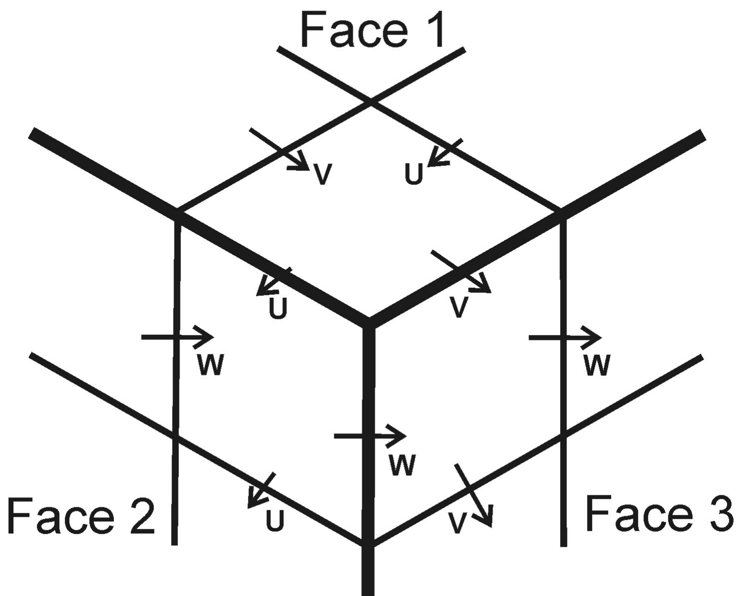
Top diagram shows projection of grid cells from the center of a cube face onto the sphere; they are squares.

Bottom diagram shows projection of grid cells form the corners of the cube onto the sphere; they are parallelograms.

This arrangement causes all kinds of terrible harmonics and numerically induced errors.

Almost any cell arrangement will cause *grid imprinting* errors, but those of an icosahedral grid is less severe than those of a cubed-sphere grid.



Icosahedral B-grid model – the Raw grid

An icosahedron has 12 vertices, and for grid level 0, these vertices are primary cell centers. Perpendicular bisecting arcs between the centers determine the primary cell edges, 12 regular pentagons for grid level 0.

Adding half way centers to the original near-by centers produces the next grid level. Grid level 1 has 12 pentagons and 30 hexagons: a soccer ball. Grid level n has $2+10*4^n$ primary cells.

Momentum cell centers are the corners of primary cells; the corners of momentum cells are the centers of primary cells. $20*4^n$ momentum cells.

The icosahedron is grouped into 5 wedges or 4 triangles each. Each wedge touches both the north and south poles. A 12 processor computer divides the sphere into a north pole cell, a south pole cell, and 10 half wedges. Coding allows the half wedges to be divided by powers of 4.

# Indexing for C cells

A four triangle wedge of hexagonal primary C cells for grid level 2 ($2^2 = 4 =$ IM cells along an icosahedron's edge).

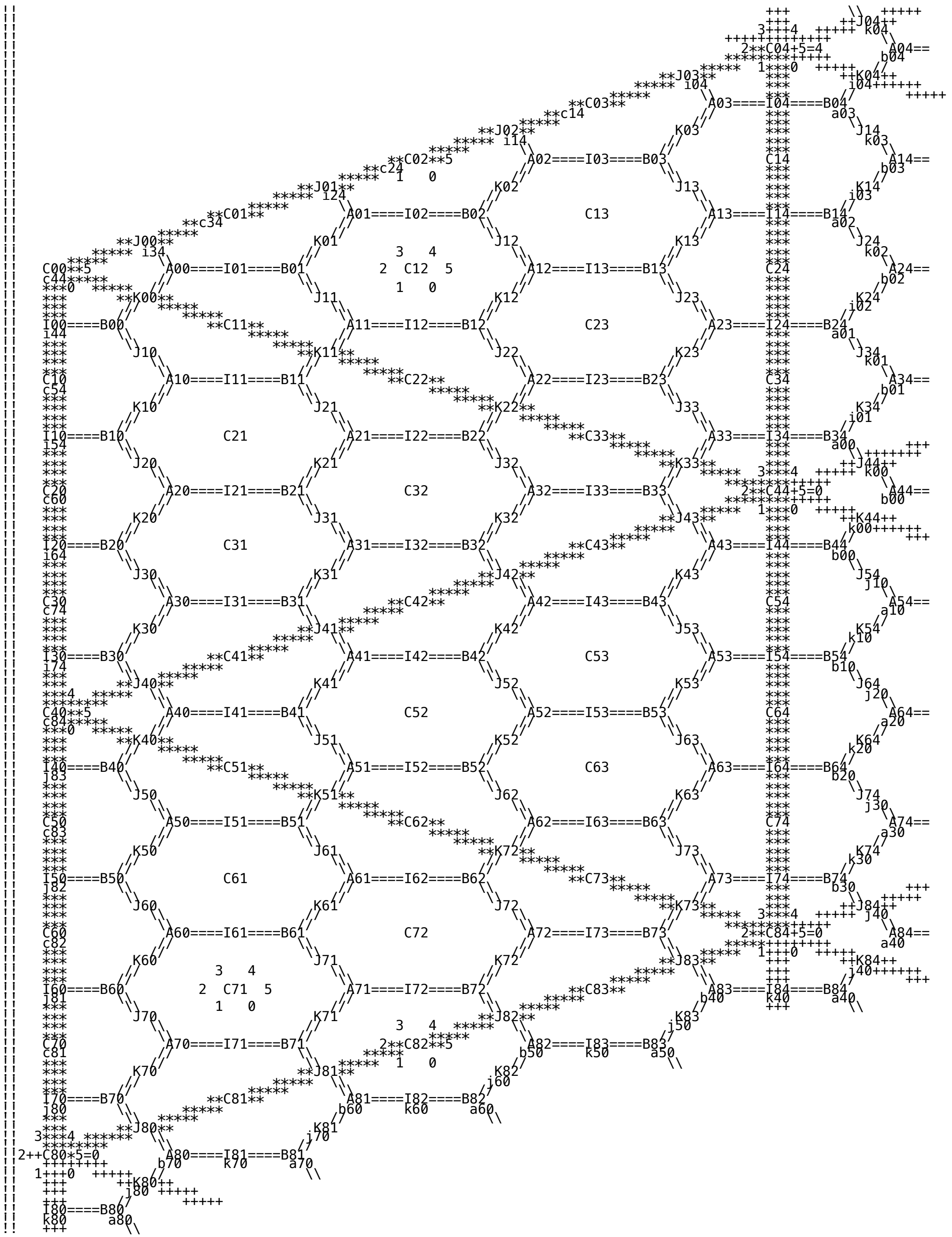
North pole is located at C04.

South pole is located at C80.

Cell C44 has same location as C00 of wedge to the east.

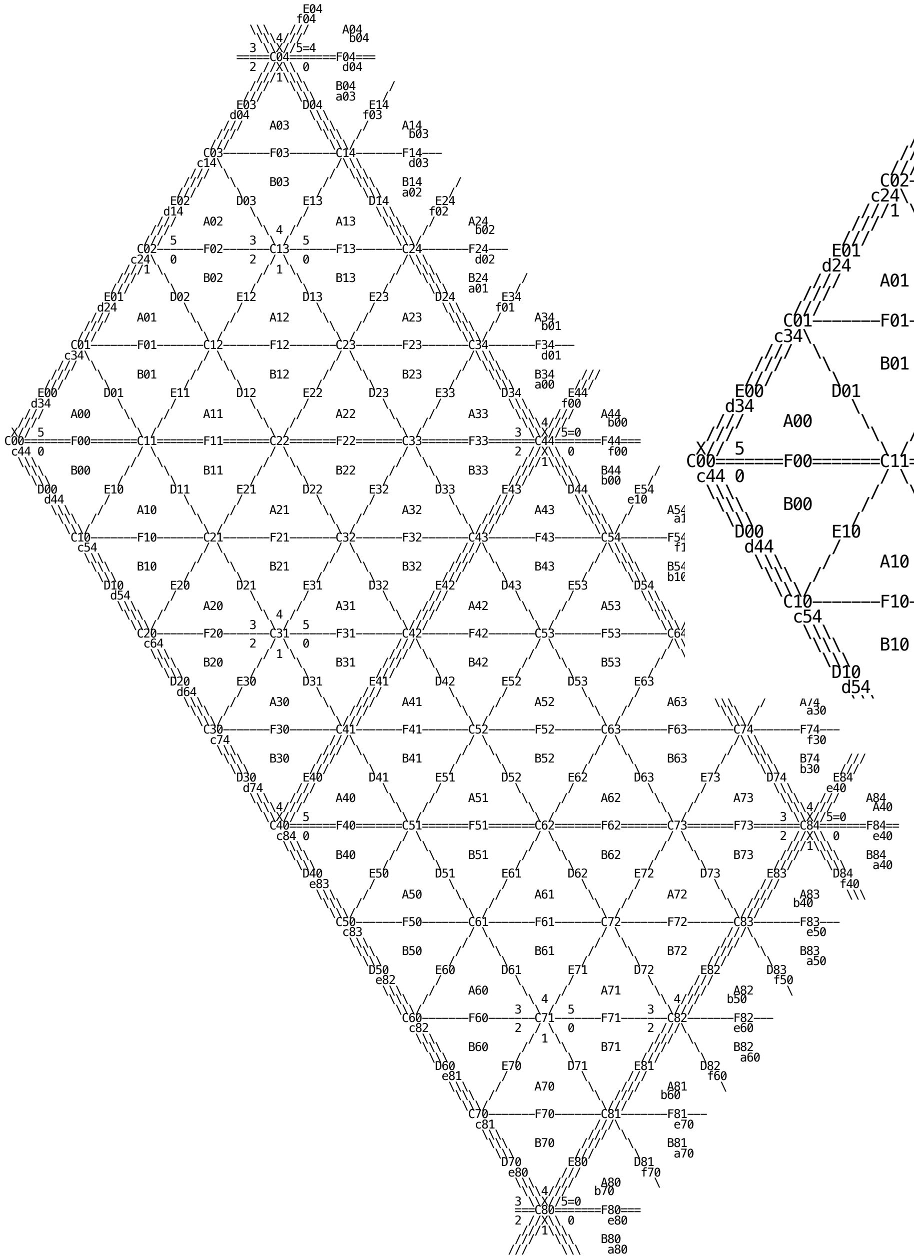
Source term computations are performed on polar cells and for C(1:IM*2, 1:IM) for each wedge.

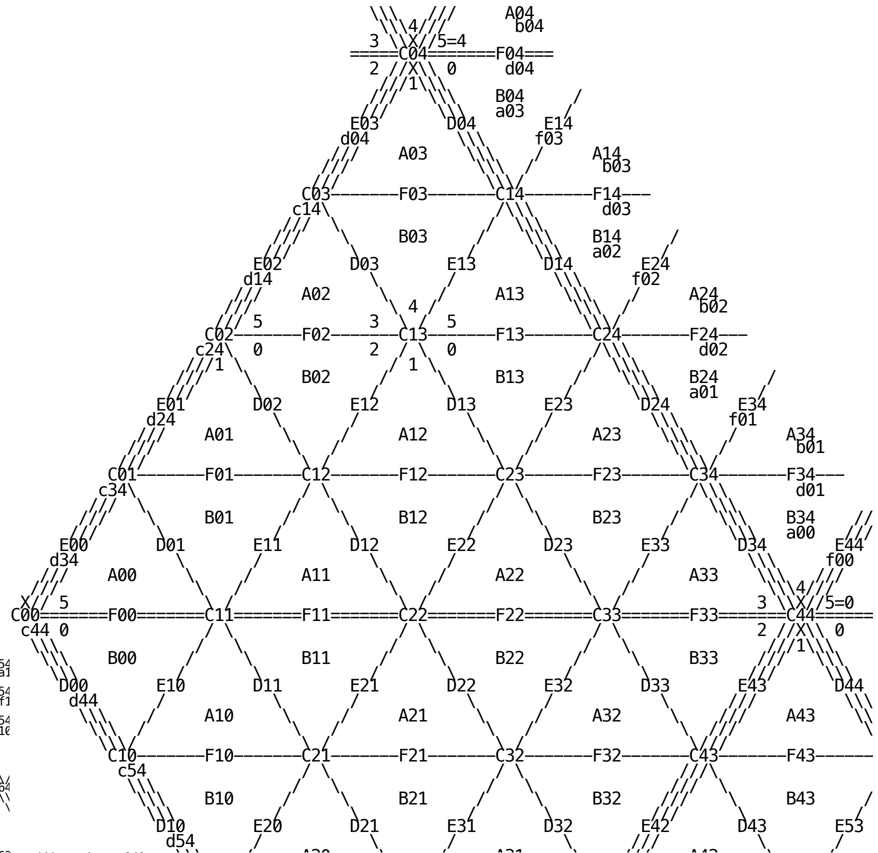
Momentum cells A and B surround all computational C cells.



<u>Indexing for A or B cells</u>

A four triangle wedge of triangular momentum A and B cells. Momentum cells are computed for A(0:IM*2-1, 0:IM-1) and for B(0:IM*2-1, 0:IM-1).



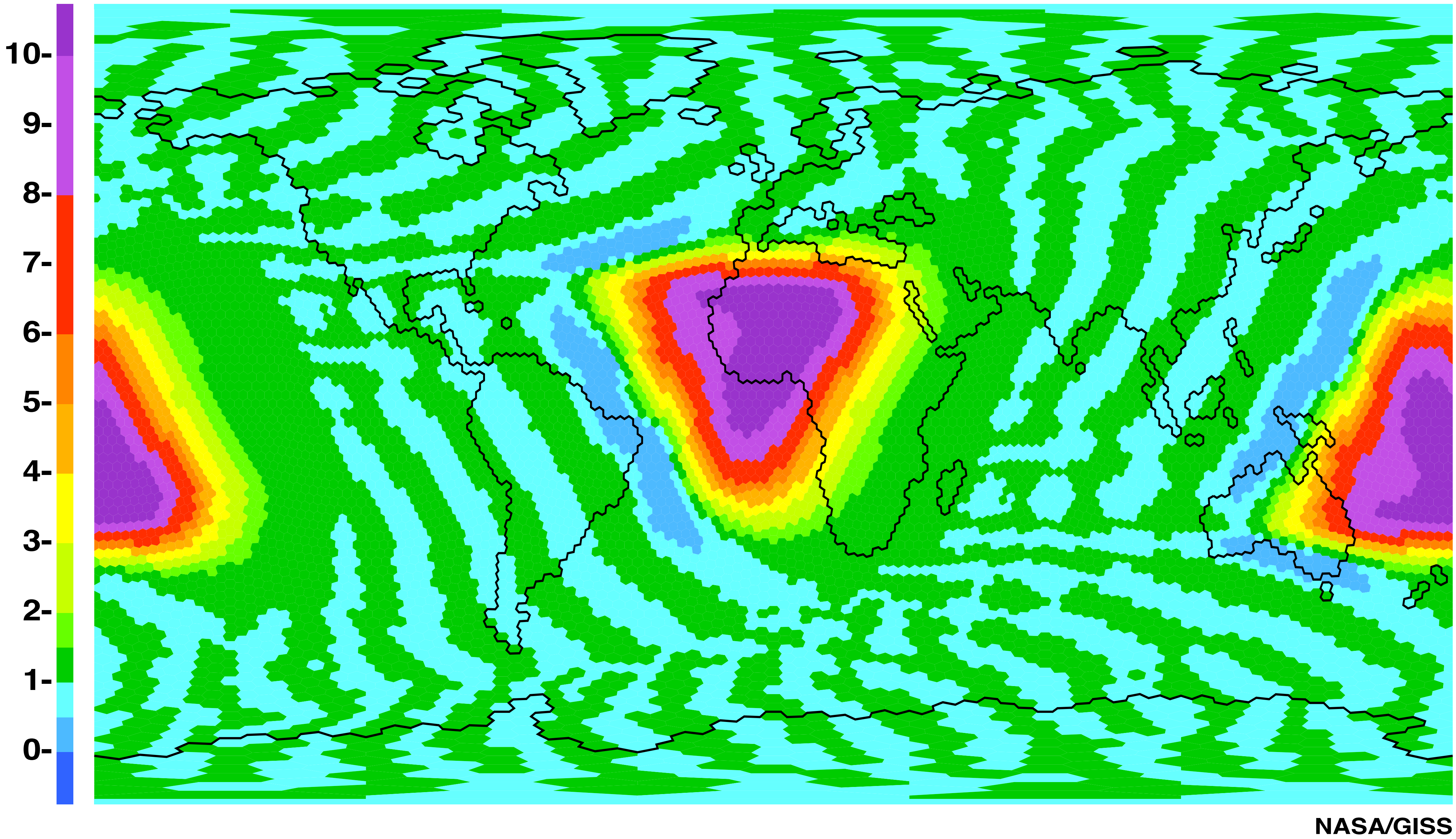


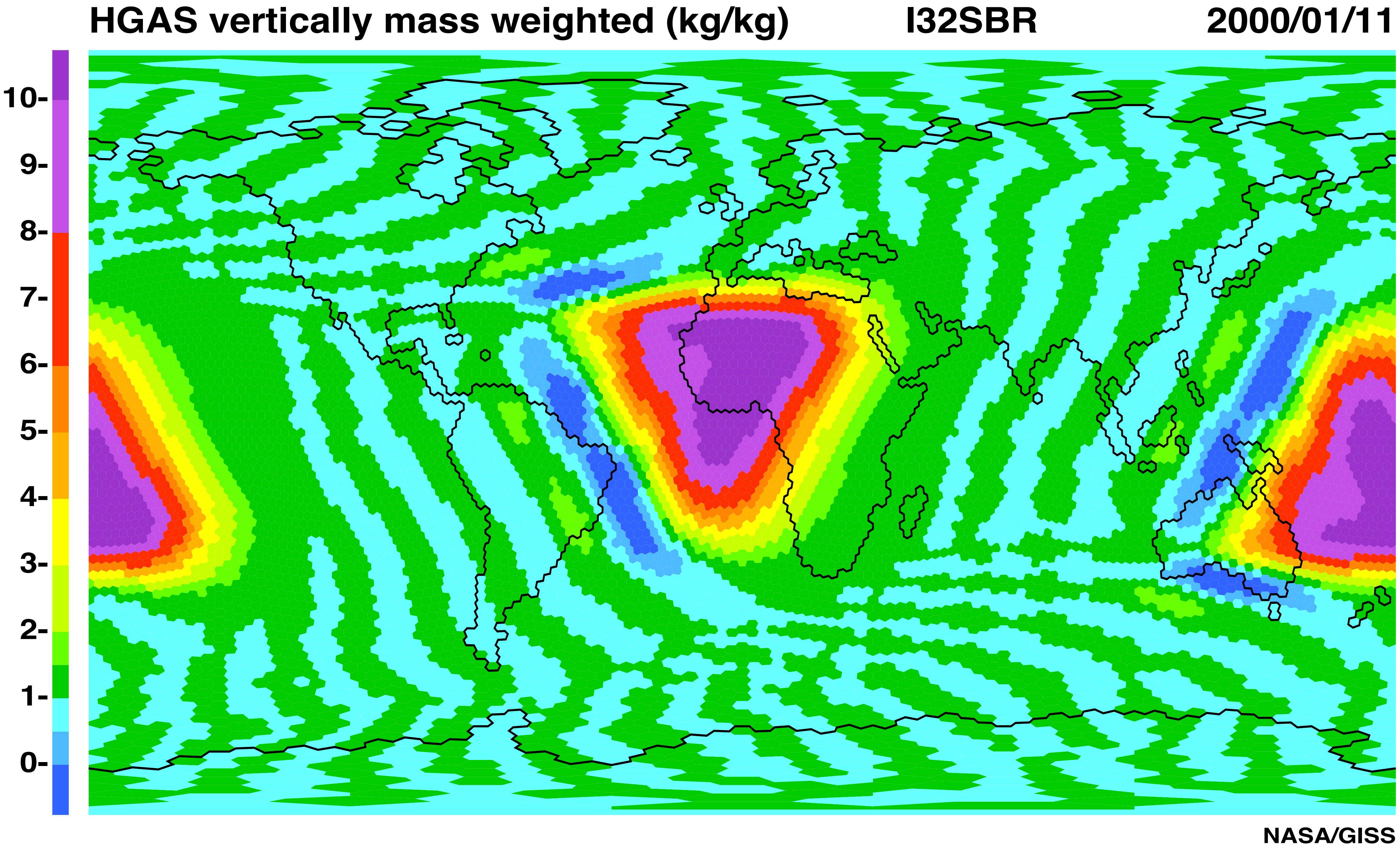
Progress to date

Starting from single layer model icosahedral B-grid model.
Momentum uses partial upstream advection; no filters are needed.
Program to produce Post-Script file on icosahedral native grid.
Multi layers are counted from top down.
Step-mountain: C cells are whole, momentum cells may be fractional.
MPI coding allows $2 + 10*4^n$ processors.
Tracer advection allows vertical gradients, but not in the horizontal.
Pressure gradient force uses 2 terms: gradients of pressure and geopotential.
Non-conservative interpolation scheme from Lat-Lon to icosahedral grid.
Z file (surface fraction and topographies) are created for icosahedral grid.
Time uses Gregorian calendar.
Atmospheric initial conditions file created and read by model.
Restart file written and read back in.
AIJ diagnostics accumulated and written to .PRT file.

Two <u>Tracer Advection Schemes</u> have been programmed.
Each scheme uses mean and vertical gradients, but no horizontal gradients.
For each advective time step: .5 vertical, full horizontal, .5 vertical.
One scheme, HGAS, has no restriction on sign or magnitude of tracers.
Second scheme, WVAP, restricts tracers to be non-negative. If summation of outgoing fluxes exceeds tracer mass in a cell, outgoing fluxes are reduced proportionally.
Schemes are tested using solid body rotation without topography; equatorial velocity is circumference divided by 10 days, ~46.3 (m/s).
Except for two equatorial triangles on opposite sides of the globe, the initial value for each scheme is 1. Initial value inside the triangles is 11, at the C cell edges it is 6, and at the C cell corners it is 3. Apex of one triangle aligns with the plot's edge. Tracer is very sharp at edges.

WVAP vertically mass weighted (kg/kg) I32SBR 2000/01/11




HGAS vertically mass weighted (kg/kg)
I32SBR
2000/01/11
10-
9-
8-
7-
6-
5-
4-
3-
2-
1-
0-
NASA/GISS

The multi-layer solid body rotation simulation with two tracers, 32 vertical layers, and grid level 5 (10242 cells) was integrated on haswell nodes of the Discover super computer at Goddard Space Flight Center. The wall-clock time for a 50 day simulation is as follows:

12 processors, 1 node, 37:13 (min:sec), time ratio = 1.00

42 processors, 2 nodes, 10:15 (min:sec), time ratio = 3.63

162 processors, 6 nodes, 2:56 (min:sec), time ratio = 12.69

642 processors, 23 nodes, 1:14 (min:sec), time ratio = 30.18

Vertical advection of tracers makes no MPI Halo subroutine calls because the advection is applied to both necessary and halo cells. With 642 processors and grid level 5, each non-polar processor performs vertical computations on 25 horizontal columns. It is also possible to perform the vertical computations on 16 columns and follow that with an MPI Halo subroutine call.

# Coding for Coriolis force

I,J,K are horizontal indices.  K indicates wedge.  J and K are combined. L is vertical layer.  There are twice as many A and B momentum cells as there are primary C cells.  XYZ = position vector **P**; XYZ(3) = sinφ.

```
      Do 20 JK=JK0,JKM  ;  Do 20 I=I0,IM
      Do 20 N=1,3  ;  Nm1 = N-1  ;  If (N==1) Nm1 = 3
                      Np1 = N+1  ;  If (N==3) Np1 = 1
      Do 10 L=1,LMAA(I,JK)
      VAN(N,L,I,JK) = VAN(N,L,I,JK) + 2*DT*OMEGA*XYZA(3,I,JK) *
                      (XYZA(Nm1,I,JK)*VA(Np1,L,I,JK) - XYZA(Np1,I,JK)*VA(Nm1,L,I,JK))
      Do 20 L=1,LMBA(I,JK)
   20 VBN(N,L,I,JK) = VBN(N,L,I,JK) + 2*DT*OMEGA*XYZB(3,I,JK) *
                      (XYZB(Nm1,I,JK)*VB(Np1,L,I,JK) - XYZB(Np1,I,JK)*VB(Nm1,L,I,JK))
```

## Pressure Gradient Force

Amean = vertically mass weighted specific volume ($m^3$/kg) on C cells

AverAmean = Amean area weighted to momentum triangular A cells

Pmean = C cell pressure (Pa) at corner of momentum A cells

GZmean = vertically mass weighted altitude (m) times Gravity ($m/s^2$)

Pmean, GZmean, Amean are located at corners of momentum cells.

```
Do JK=JK0,JKM  ;  Do I=I0,IM
   Do L=1,LMAA(I,JK)
      AverAmean = FSADL(L,I,JK)*Amean(L,I,JK) + FSAEL(L,I,JK)*Amean(L,I+1,JK+1) + FSAFL(L,I,JK)*Amean(L,I,JK+1)
      VA(:,L,I,JK) = VA(:,L,I,JK) + DT * &
                     (xGZC00A(:,L,I,JK) * (AverAmean*Pmean(L,I  ,JK  ) + GZmean(L,I  ,JK  )) + &
                      xGZC11A(:,L,I,JK) * (AverAmean*Pmean(L,I+1,JK+1) + GZmean(L,I+1,JK+1)) + &
                      xGZC01A(:,L,I,JK) * (AverAmean*Pmean(L,I  ,JK+1) + GZmean(L,I  ,JK+1)))  ;  EndDo
```

## Advection of momentum

`MFLUXA` and `VFLUXA` are vertical fluxes of mass and momentum.

`FLUXD` , `FLUXE` , `FLUXF` are horizontal fluxes on momentum cell edges.

`yAREAAL`($m^{-2}$) = reciprocal of area for momentum cell A.

```
!**** Advect mass and angular momentum on A cells of Layers 2:LM-1
      Do 520 L=2,LM-1
      VAN(:,L,I,JK) = VAO(:,L,I,JK) * MAO(L,I,JK) + DT * yAREAAL(L,I,JK) * &
                      (VFLUXA(:,L-1) - VFLUXA(:,L) - VFLUXD(:,L,I,JK+1) - VFLUXE(:,L,I,JK) - VFLUXF(:,L,I,JK))
      MAN(  L,I,JK) = MAO(  L,I,JK)               + DT * yAREAAL(L,I,JK) * &
                      (MFLUXA(  L-1) - MFLUXA(  L) - MFLUXD(  L,I,JK+1) - MFLUXE(  L,I,JK) - MFLUXF(  L,I,JK))
  520 VAN(:,L,I,JK) = VAN(:,L,I,JK) / MAN(L,I,JK)
```

Future Work for Model-I

Year 1: Source term subroutines from older Atmosphere-Ocean Model (AOM) will be re-indexed to work on symmetric icosahedral grid. Some variables will be renamed. Resulting Model-I should work as well as AOM did, but numerous features will be similar to Model-E.

Years 2-5: Model-E coding will be converted to Fortran-90 and cleaned up with proper indentation. Some features of Model-I (Gregorian calendar, water vapor mass, total energy conservation per subroutine) will be implemented into Model-E. One at a time, Model-E subroutines will be re-indexed and be inserted into Model-I.